IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Patent Application No. 10/059,427

Applicant: Leijten

Filed: January 29, 2002

TC/AU: 2183

Examiner: Petranek, Jacob Andrew

Docket No.: 260695 (Client Reference No. P82094US00)

Customer No.: 23460

RESPONSE TO ORDER RETURNING
UNDOCKETED APPEAL TO EXAMINER

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the communication dated November 27, 2007 in the above-mentioned case, a concise explanation of the subject-matter in each of the independent and dependent claims is provided, as required according to rule § 41.37 (c). The concise explanation of the subject-matter is provided below in the form of wording indicated in italic inserted between the wording of the latest submitted version of the claims. The source of the explanatory wording in the description and the drawing is indicated.

Summary of Claimed Subject Matter

The summaries of the claims reference the specification and drawings filed with the application on January 29, 2002.

Appellants invention recited in **claim 1** is directed to a computer system with a processing unit (32, 34) and a memory (30). Figure 3 schematically shows the computer system comprising an instruction processing unit (34) and a memory (30). The processing unit (34) is arranged to fetch memory lines from the memory (30) and execute instructions from the memory lines. In this regard, Figure 1 shows a VL^2IW instruction for execution by the computer system of Figure 3. As shown therein the instructions comprise a header 1 and at most C operation segments. The instructions are stored in the memory (30) as illustrated in Figure 2. The processing unit (32, 34) is capable of fetching memory lines from the memory (30), in that it comprises an instruction issue unit (32).The instruction issue unit (32) fetches memory lines from memory, decompresses instructions from the memory lines and sends the instructions to the instruction processing unit (34) for execution. (p. 7, lines 13-20)

Each memory line is fetched as a whole and is capable of holding more than one instruction. In this regard the application discloses an instruction issue unit 32, addressing unit 326 outputs addresses of memory lines to memory 30. Each memory line (i.e. data from a series of e.g. 8 locations addressed by an address from the addressing unit 326) output from the memory is applied to the instruction selection units 322a-c. (p. 7, lines 21-30) As illustrated in Figure 2 and described in page 9, lines 29 et seq., a memory line may comprise more than one instruction (e.g. the second line has three instructions, the third line comprises two instructions and part of a third instruction).

Claim 1 further recites that at least one instruction comprises information, inserted at compile time, that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line. In this regard the application discloses in Figure 1, instructions comprising a header (1) with information in the form of fetch control bits F (p. 9, lines 25-28). According to the present invention three fetch control bits are defined. These are the stall (S) bit, the realign (R) bit, and the prefetch (P) bit. These are shown in detail in Figure 2. The prefetch bit is always present. It indicates to the instruction fetch logic of the processor that the next VL^2IW instruction to be fetched does not fit on the current memory line, such that the next memory line must be fetched in order to enable decompression and

execution of the next instruction. Using this bit eliminates the need for the hardware to determine the right moment of prefetching. The right moment is the moment at which unnecessary fetching is minimized, for example to reduce power dissipation. If the prefetch bit is not used an additional costly calculation on the critical path of the instruction fetch hardware is required. (p. 11, lines 10-18) Depending on the requirements of the processor instance, either the stall bit, the realign bit, or both are used in the instruction format. The realign bit indicates to the instruction expansion logic that the next instruction to be decompressed is realigned. That is, the padding bits following the current instruction should be skipped. On a taken branch, the stall bit indicates to the instruction fetch logic that the instruction being branched to should be fetched in two instead of one cycle, i.e. that a stall cycle is required. (p. 11, lines 19-24)

Finally, claim 1 recites that the processing unit is arranged to respond to the information by controlling said part as signaled by the information. In this regard, paragraphs 36-38 describe how the processing unit is arranged to respond to this information. The processing unit controls the addressing unit (326) in response to the prefetch bit P (p. 8, lines 13-22). The processing unit controls the selection control unit (324) in response to the realign bit (R). The processing unit stalls execution (which is carried out by instruction processing unit 34) in response to the stall bit (S).

Regarding **claim 8**, a method of processing instructions in a computer system with a processing unit and a memory is claimed. In this regard, Figure 3 schematically shows the computer system. The system comprises an instruction processing unit (34) and a memory (30) wherein the processing unit is arranged to fetch memory lines from the memory Figure 1 shows a VL^2IW instruction for execution by the computer system of Figure 3. As shown therein the instructions comprise a header 1 and at most C operation segments. The instructions are stored in the memory (30) as illustrated in Figure 2. The processing unit (32, 34) is capable of fetching memory lines from the memory (30), in that it comprises an instruction issue unit (32). The instruction issue unit (32) fetches memory lines from memory, decompresses instructions from the memory lines and sends the instructions to the instruction processing unit (34) for execution. (p. 7, lines 13-20)

Claim 8 furthermore recites executing instructions from the memory lines capable of holding more than one instruction. Referring to the specification and drawings, in the instruction issue unit 32, the addressing unit 326 outputs addresses of memory lines to memory 30. Each memory line (i.e., data from a series of e.g. 8 locations addressed by an address from the addressing unit 326) output from the memory is applied to the instruction

selection units 322a-c. (page 7, lines 21-30) As illustrated in Figure 2 and described in page 9, lines 29 et seq., a memory line may comprise more than one instruction (e.g. the second line has three instructions, the third line comprises two instructions and part of a third instruction).

Claim 8 further recites at least one instruction comprising information, inserted at compile time, that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line. As illustrated in Figure 1, the instructions comprise a header (1) with information in the form of fetch control bits F (p. 9, lines 25-28). According to the present invention three fetch control bits are defined. These are the stall (S) bit, the realign (R) bit, and the prefetch (P) bit. These are shown in detail in Figure 2. The prefetch bit is always present. It indicates to the instruction fetch logic of the processor that the next VL^2IW instruction to be fetched does not fit on the current memory line, such that the next memory line must be fetched in order to enable decompression and execution of the next instruction. Using this bit eliminates the need for the hardware to determine the right moment of prefetching. The right moment is the moment at which unnecessary fetching is minimized, for example to reduce power dissipation. If the prefetch bit is not used an additional costly calculation on the critical path of the instruction fetch hardware is required. (p. 11, lines 10-18) Dependent on the requirements of the processor instance, the stall bit, the realign bit, or both are used in the instruction format. The realign bit indicates to the instruction expansion logic that the next instruction to be decompressed is realigned, that is, the padding bits following the current instruction should be skipped. On a taken branch, the stall bit indicates to the instruction fetch logic that the instruction being branched to should be fetched in two instead of one cycle, i.e. that a stall cycle is required. (page 11, lines 19-24)

Claim 8 further recites that the method comprises fetching each memory line as a whole and processing an instruction from a current memory line. This corresponds to the specifications' description of the instruction issue unit (32) fetching memory lines from memory, decompressing instructions from the memory lines and sending the instructions to the instruction processing unit (34) for execution. (page 7, lines 13-20)

Claim 8 further recites reading the information from the instruction during processing. This corresponds to header selection unit 320 supplying information from the header of an instruction in a memory line received from memory 30 or previous line register 328. (p. 7, line 31 to p. 8, line 12)

Finally, claim 8 recites controlling said part as signaled by the information. The corresponding disclosure states that the header in header selection unit 320 supplies additional information from the header that simplifies the hardware needed for fetching of memory lines that contain subsequent instructions. A first bit from the header selection unit 320 is the "prefetch (P) bit". This bit indicates whether the next instruction to be executed is contained in the memory line presently received from memory 30, or whether part or whole of the instruction is contained in a next memory line. In the latter case, addressing unit 326 responds to the prefetch bit by issuing a prefetch command for that next memory line to memory 30. When the next memory line is prefetched, the old memory is copied into the previous line register 328 so that the part of the instruction that is in the previous line can be accessed from there by the instruction selection units 322a-c. (p. 8, lines 13-22) A second bit from the header selection unit 320 is the "realign (R) bit" the realign bit indicates whether the instruction is followed by padding, which should not be executed. If so, the selection control unit 324 will respond to the realignment bit by updating the position indicated to the instruction selection units 322a-c so that this position will point to the start of the next memory line for the next instruction. (p. 8, lines 23-27) A third bit is from the header selection unit 320 the "stall (S) bit". The stall bit indicates whether the present instruction is spread over more than one line in case the last of those lines has not yet been fetched. This will happen typically in case the instruction is a branch target that is spread over more than one line so as to avoid loss of memory space due to realignment. In this case, the instruction processing unit 32 will respond to the stall bit by stalling execution for the period needed to fetch the next memory line that contains the remainder of the instruction. (p. 8, lines 28-34)

Respectfully submitted,

Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza
180 North Stetson Ave., Suite 4900
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: January 28, 2008